

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Studi Literatur

Penelitian ini dilakukan berdasarkan beberapa penelitian terdahulu yang sejenis. Pada bab ini, akan menuliskan mengenai rincian penelitian sejenis yang telah dilakukan sebelumnya. Hasil dari beberapa penelitian terdahulu yang menjadi acuan dalam penelitian ini telah ditulis pada Tabel 1.

Tabel 1. Penelitian Terdahulu yang Sejenis

Judul (Penulis, Tahun)	Dataset	Metode	Hasil			
			AUC	ACC	Precision	Recall
Microsatellite Instability Prediction Of Uterine Corpus Endometrial Carcinoma Based On H&E Histology Whole-Slide Imaging (T. Wang et al., 2020)	UCEC Cohort dengan 516 pasien yang didapatkan dari TCGA.	ResNet-18 with PALHI	0.73	-	-	-
Clinical-grade Detection of Microsatellite Instability in Colorectal Tumors by Deep Learning (Echle et al., 2020)	TCGA 5000 data pasien kanker usus stadium I-IV di Amerika Serikat	Modified ShuffleNet Deep Learning	0.92	-	67%	95%
Deep learning for prediction of colorectal cancer outcome: a discovery and validation study (Skrede et al., 2020)	Data pasien kanker usus berjumlah 920 di UK dan 1122 di Norway	CNN Mobilenet V2	0.71	76%	78%	52%

A novel approach combined transfer learning and deep learning to predict TMB from histology image  (L. Wang et al., 2020)	Data pasien kanker lambung dan usus berjumlah 217.916 dan 154.063	CNN VGG19	0.75 STAD dan 0.82 COAD	71% STAD dan 77% COAD	-	-
---	---	-----------	-------------------------	-----------------------	---	---

## 2.2. *Microsatellite Instability*

Mikrosatelit tidak stabil (MSI) adalah kondisi yang disebabkan oleh kesalahan mismatch repair (MMR) DNA. MSI sering ditemukan pada kanker *gastrointestinal* (pencernaan) dengan kemungkinan mengalami *Lynch Syndrome*. Deteksi MSI dilakukan menggunakan *immunohistochemistry* (IHC) dan *polymerase chain reaction* (PCR). Pasien dengan status MSI yang tinggi menunjukkan prognosis yang baik dengan asumsi limfosit yang protektif terhadap MSI sehingga metastasis sel kanker. Selain itu, pasien dengan status MSI tidak mendapatkan pengaruh dari kemoterapi *5-fluorouracil adjuvant*. [14].

## 2.3. *Augmentasi Data*

Tahap augmentasi dilakukan untuk meningkatkan jumlah data dengan cara membuat data baru dengan memodifikasi data yang ada hingga program mengenali data tersebut adalah data yang berbeda. Hal tersebut membuat data memiliki lebih banyak variabilitas yang dapat memudahkan model melakukan proses klasifikasi terhadap objek [15]. Tahap augmentasi ini dilakukan pada kasus dengan jumlah data sedikit untuk mengurangi overfitting dengan beberapa teknik transformasi seperti rotasi, perbesaran, pegeseran dan sebagainya [16], [17].

## 2.4. *Convolutional Neural Networks*

*Convolutional Neural Networks* merupakan metode dari salah satu kelas *deep learning* yang dapat digunakan dalam permasalahan pemrosesan gambar [18]. CNN bekerja seperti persepsi visual pada mata manusia yang mengenali fitur rumit atau pola pada sebuah gambar secara bertahap sedikit demi sedikit. Metode CNN lebih

baik dibandingkan metode lain seperti ANN meski memiliki jumlah lapisan yang sama. Hal tersebut disebabkan oleh filter konvolusional yang dapat mengekstraksi fitur tingkat tinggi dari gambar. CNN memiliki beberapa layer seperti *input layer*, *convolutional layer*, *activation layer*, *batch normalization layer*, *pooling layer*, *dropout layer*, *fully connected layer* dan *output layer* [19].

#### 2.4.1. Input Layer

Pada dasarnya CNN memiliki *input layer*, *output layer* dan beberapa *hidden layer*. *Input layer* merupakan lapisan pertama yang akan memproses gambar dalam CNN. Pada input layer terdapat parameter untuk deklarasi ukuran gambar dan format gambar. Gambar yang masuk dapat memiliki bermacam-macam ukuran tanpa ada batasan dengan format *grayscale* atau berwarna (RGB) [19].

#### 2.4.2. Convolutional Layer

Lapisan konvolusi adalah lapisan berikutnya setelah input layer. Lapisan ini terdiri dari neuron yang terhubung ke area kecil *input* gambar atau filter dengan ukuran (3x3), (5x5), (7x7), dll. Lapisan konvolusi juga memiliki parameter lain sebagai optimasi seperti padding untuk menentukan jumlah pixel bernilai 0 yang akan ditambahkan di setiap sisi dari input, kemudian parameter stride digunakan untuk menentukan berapa pixel yang akan dilewati saat proses konvolusi [20]. Proses konvolusi dapat dilihat pada Gambar 1 [21].

Image					Convolved Feature		
1	1	1	0	0	1	1	1
0	1	1	1	0	0	1	1
0	0	1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0	1
0	0	1 <sub>x0</sub>	1 <sub>x1</sub>	0 <sub>x0</sub>			
0	1	1 <sub>x1</sub>	0 <sub>x0</sub>	0 <sub>x1</sub>			

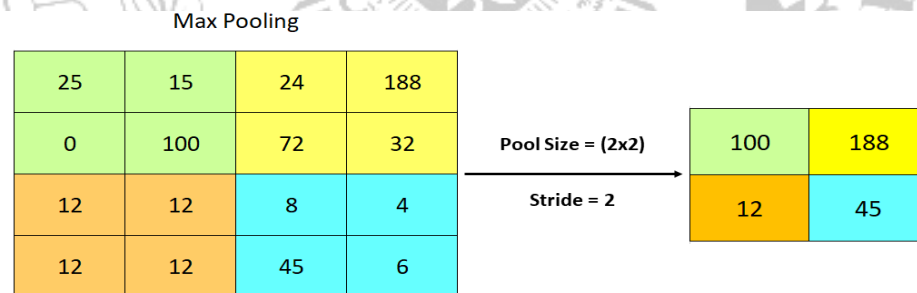
Gambar 1. Proses Konvolusi

### 2.4.3. Batch Normalization Layer

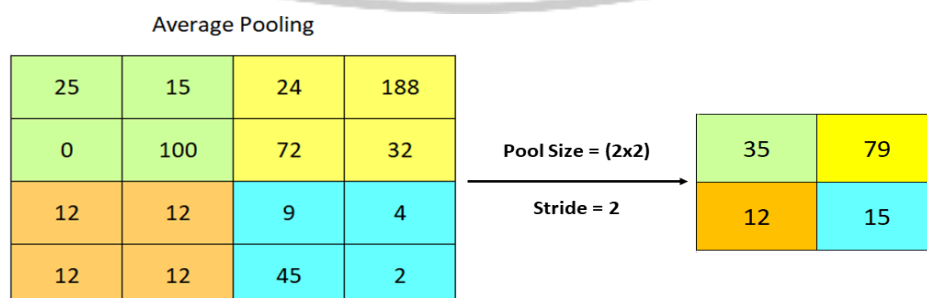
*Batch Normalization* merupakan lapisan yang dapat mengurangi pergeseran kovarian. Metode ini dapat meningkatkan kecepatan pelatihan dan akurasi pelatihan secara signifikan [22]. Selain itu, lapisan *batch normalization* dapat mengatasi overfitting. Beberapa penelitian mendapatkan hasil lebih baik dengan hanya menggunakan lapisan *batch normalization* tanpa menggunakan *dropout* [23].

### 2.4.4. Pooling Layer

Lapisan pooling digunakan untuk meningkatkan efisiensi komputasi dengan mengurangi dimensi dari peta fitur namun tetap mempertahankan informasi penting didalamnya [24]. Proses yang dilakukan pada lapisan ini dipengaruhi oleh parameter *pool size* dan *stride* sama seperti lapisan konvolusi. Teknik pada lapisan pooling yang paling umum digunakan adalah *max pooling* karena kecepatan *average pooling*. *Max pooling* mengambil nilai tertinggi dari area yang dipilih, hal tersebut membuat *max pooling* lebih efisien sedangkan *average pooling* mengambil nilai rata-rata dari area sehingga menyebabkan teknik ini lebih kompleks. Proses *max pooling* dan *average pooling* dapat dilihat pada Gambar 2 dan Gambar 3 [25].



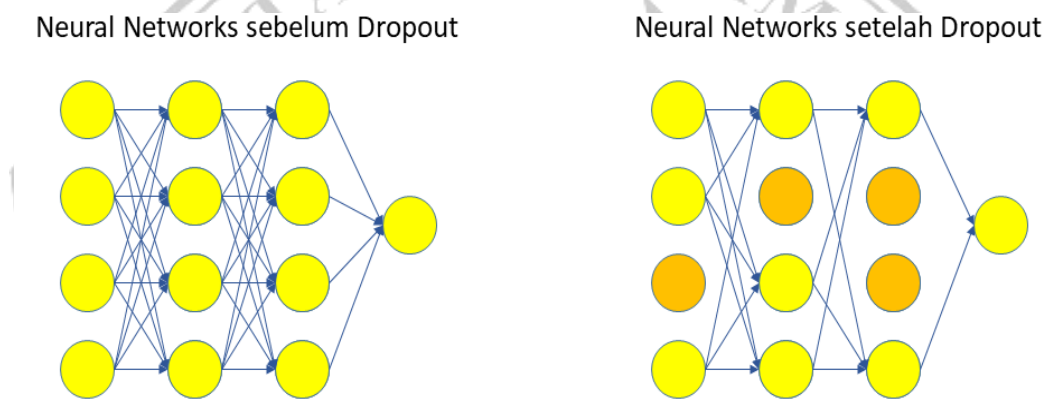
Gambar 2. Max Pooling



Gambar 3. Average Pooling

#### 2.4.5. Dropout Layer

*Dropout* merupakan teknik regularisasi paling sederhana dan baik dalam kinerja untuk mengurangi permasalahan overfitting selain teknik augmentasi data dan *batch normalization* [26]. Pada umumnya, *dropout* terletak pada lapisan *fully connected layer*. *Dropout* bekerja dengan cara memilih beberapa neuron dalam lapisan *fully connected layer* secara acak dan mengubah nilai neuron tersebut menjadi 0. Selain pada *fully connected layer* penambahan lapisan dropout juga diletakkan pada setelah pooling layer [27]. Pada beberapa kasus, dropout dapat mengatasi overfitting dengan mengurangi jarak antara akurasi dari data training dan validasi pada saat pelatihan [28]. Proses dropout dapat dilihat pada Gambar 4 [29].



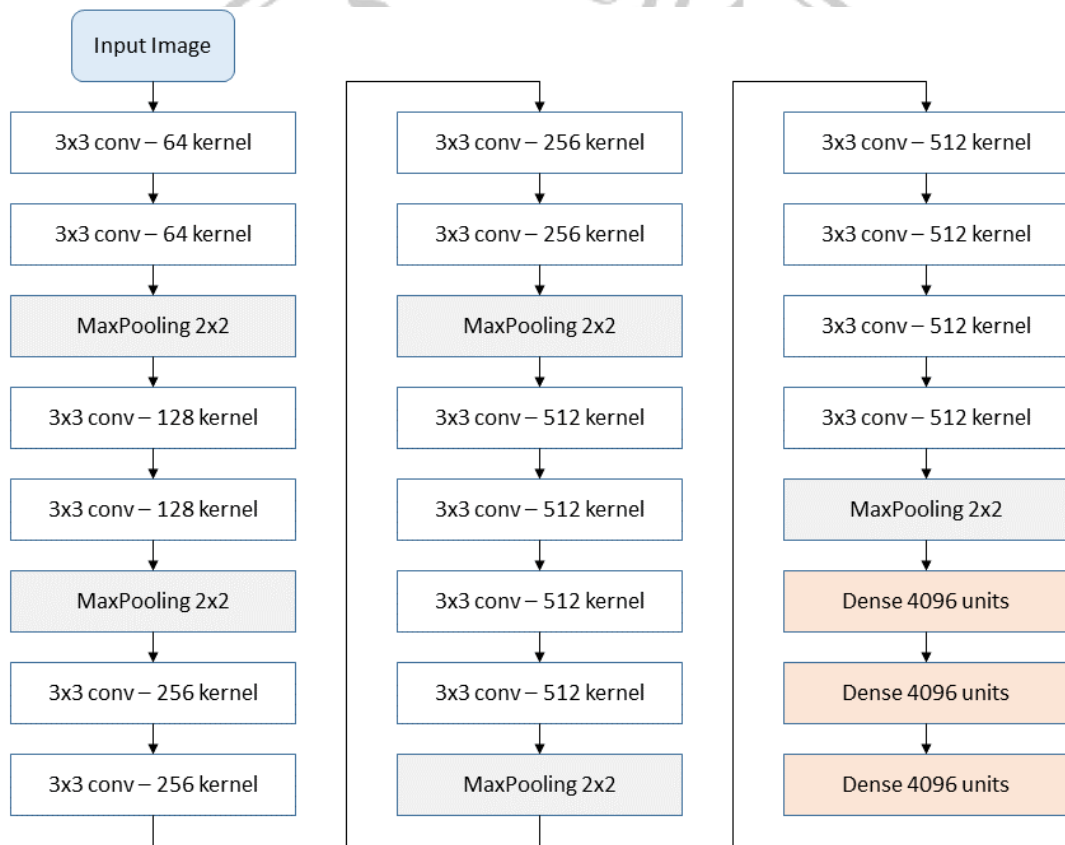
Gambar 4. Proses Dropout

#### 2.4.6. Fully Connected Layer

Lapisan terakhir dari *Convolutional Neural Networks* adalah *Fully Connected Layer* [30]. Terdapat beberapa lapisan pada *fully connected layer* seperti *dropout layer*, *hidden layer*, dan *output layer*. Lapisan ini dapat mengubah *feature map* dengan bentuk vektor multi dimensi menjadi vektor satu dimensi melalui proses flaten [31]. Vektor tersebut akan digunakan sebagai input untuk lapisan *hidden layer* dan *output layer*. Selain itu, lapisan juga terdapat fungsi aktivasi berupa *sigmoid* untuk data yang memiliki satu kelas dan *softmax* untuk data multi kelas.

## 2.5.VGG19

VGG19 adalah arsitektur jaringan yang merupakan bentuk variasi dari VGG16 yang diciptakan oleh Visual Geometry Group (VGG) di Universitas Oxford pada tahun 2014 dan menjadi terkenal karena mendapatkan performa yang baik dalam beberapa kasus klasifikasi gambar. Model ini memiliki 16 lapisan konvolusi dengan filter 3x3, kemudian terdapat 5 *pooling layer* yang menggunakan MaxPooling dengan *pool size* sebesar 2x2 dan 3 fully connected layer dengan nilai 4096, 4096, 1000 lalu size image untuk input memiliki ukuran 224 x 224 piksel [32]. Detail struktur model VGG19 dapat dilihat pada Gambar 5 [33].



Gambar 5. Struktur Model VGG19

## 2.6. Pengujian Klasifikasi Model

Tahap ini merupakan tahap akhir untuk menilai performa dari model berdasarkan hasil pelatihan yang telah dilakukan sebelumnya terhadap dataset. Penilaian ini dilakukan dengan menggunakan *confusion matrix* yang kemudian

digunakan untuk mencari nilai parameter seperti akurasi, presisi, recall, f1-score dan AUC (*Area Under Curve*). *Confusion matrix* akan dijabarkan dalam bentuk tabel dengan komposisi seperti pada Tabel 2 [34].

Tabel 2. Confusion Matrix

		True Value	
		Positive	Negative
Predict Value	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Berdasarkan confusion matrix pada tabel diatas, perhitungan nilai lainnya dapat dilakukan dengan mengambil nilai TP, FP, FN, dan TN. Detail mengenai penjelasan dan perhitungan parameter lainnya sebagai berikut:

- a. **Akurasi**, merupakan perbandingan antara prediksi yang benar baik itu positif maupun negatif terhadap keseluruhan data selama pengujian. Rumus dari akurasi dapat dilihat pada persamaan berikut:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

- b. **Presisi**, merupakan perbandingan antara prediksi benar positif terhadap seluruh prediksi positif. Presisi dapat dikatakan sebagai ketepatan dari model dalam memprediksi. Rumus dari presisi dapat dilihat pada persamaan berikut:

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

- c. **Recall**, atau sensitivitas merupakan perbandingan dari prediksi benar positif terhadap seluruh data yang bernilai benar positif. Recall dapat disebut sebagai kemampuan model dalam mengidentifikasi data yang relevan. Rumus dari recall dapat dilihat pada persamaan berikut:



$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

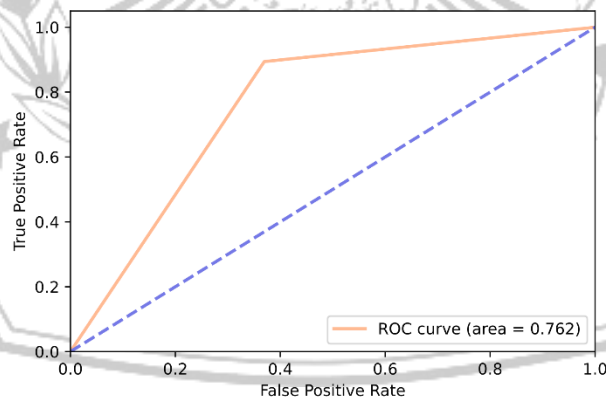
- d. **F1-Score**, merupakan perbandingan antara rata-rata nilai presisi dan recall yang didapatkan. Rumus dari F1-Score dapat dilihat pada persamaan berikut:

$$F1 - Score = 2 \frac{(Precision)(Recall)}{(Precision + Recall)} = \frac{2TP}{(2TP + FP + FN)} \quad (4)$$

- e. **AUC-ROC**, atau (*Area Under Curve*) - (*Receiver Operating Characteristics*) adalah cara untuk mengevaluasi kinerja model. Grafik AUC-ROC didapatkan dengan melibatkan TPR (*True Positive Rate*) dan FPR (*False Positive Rate*). Ilustrasi grafik AUC-ROC dapat dilihat pada Gambar 6, sedangkan rumus dari TPR dan FPR dapat dilihat pada persamaan berikut:

$$TPR = \frac{TP}{(TP + FN)} \quad (5)$$

$$FPR = \frac{FP}{(TN + FP)} \quad (6)$$



Gambar 6. Grafik AUC-ROC